06

Haga lo que hicimos en aula: performance de consultas

Al construir nuestra aplicación mediante el uso de recursos que realizan operaciones que no se encuentran explicitas debemos estudiar la documentación para entender cual es la ciencia detrás del framework o recurso.

Cuando realizamos consultas con la anotación `@ManyToOne` o `@OneToOne` detrás de escena JPA aplica una estrategia de cargamento de información llamada Eager o Anticipada o Proactiva realizando JOINS entre tablas. Pero no acaba allí, ya que si esa entidad que tiene el JOIN, tiene otras entidades dentro de sus atributos marcados con la anotación finalizando ToOne, también serán cargadas dentro de la consulta. Esto puede saturar la memoria y afectar seriamente la velocidad de carga, ya para corregir debemos utilizar el parámetro de carga FerchType.LAZY en todas las anotaciones ToOne que le indica a JPA solo cargar la entidad si es solicitada.

```
public class Producto{
…
    @ManyToOne(fetch=FetchType.LAZY)
    private Categoria categoria;
```

COPIA EL CÓDIGO

```
public class Pedido {
    …
    @ManyToOne(fetch=FetchType.LAZY)
    private Cliente cliente;
```

COPIA EL CÓDIGO

```
public class ItemsPedido {
    …
    @ManyToOne(fetch=FetchType.LAZY)
    private Producto producto;

    @ManyToOne(fetch=FetchType.LAZY)
    private Pedido pedido;
```

COPIA EL CÓDIGO

- Al realizar esta corrección se presenta un posible inconveniente donde nos encontremos con la necesidad de utilizar ese atributo de entidad. Pero para ese momento ya el EntityManager se puede encontrar cerrado, por lo que tenemos que planear nuestras consultas previniendo, el uso de esa entidad aún cuando se encuentre cerrado el EntityManager.

```
public Pedido consultarPedidoConCliente(Long id) {
    String jpql="SELECT p FROM Pedido p JOIN FETCH p.cliente WI
    return em.createQuery(jpql,Pedido.class).setParameter("id"
}

public class PruebaDeDesempenho {
    public static void main(String[] args) throws FileNotFoundI
//        LoadRecords.cargarRegistros();
        EntityManager em = JPAUtils.getEntityManager();
        PedidoDao pedidoDao = new PedidoDao(em);
        Pedido pedidoConCliente = pedidoDao.consultarPedidoCon(
        em.close();

//        System.out.println(pedido.getFecha());
//        System.out.println(pedido.getItems().size());
        System.out.println(pedidoConCliente.getCliente().getNo
```

```
        }
    }
```

COPIA EL CÓDIGO